# A SUN TRACKER FOR PLANETARY ANALOG ROVERS

**Matthew C. Deans[1], David Wettergreen[2], and Daniel Villa[2]**

[1]NASA Ames Research Center, QSS Group Inc., Moffett Field CA, USA
[2]Carnegie Mellon University, Pittsburgh PA, USA

## ABSTRACT

This paper describes the principles, design, implementation, use, and performance of a sun tracker for fixed reference orientation estimation. With relatively simple, familiar, inexpensive and low power off-the-shelf components and straightfoward modeling and calibration, a sun tracker can provide full 3-DOF orientation with accuracy well within a degree of roll pitch and yaw, and without drift. This can enable high precision long distance navigation in a Mars relevant fashion, i.e. without use of physical properties such as Earth's magnetosphere or modern infrastructure such as GPS. Most importantly, the heading errors are fixed over time, unlike estimates derived from dead reckoning or integration of inertial rate sensors.

Key words: Sun tracker, sun sensor, camera calibration, position estimation.

## 1. INTRODUCTION

Celestial navigation, using measurements of the sun, moon, and stars to estimate position, is an ancient technique. Devices such as the kamal, astrolabe, octant, and sextant, and the geometric methods to interpret measurements have guided explorers for hundreds or thousands of years[3]. Today, star trackers are used to estimate orientation of spacecraft and sun trackers are often used to estimate orientation of planetary rovers such as the MER rovers currently on Mars.

Our application, the Life in the Atacama (LITA) rover field campaign[12], requires Mars relevant position estimation with the desiderata that error estimates are on the order of 5% of distance travelled, which in turn requires heading accuracy of $3^o$. Drift over time is unacceptable. Magnetic compasses do not work on Mars, and integration of rate sensors quickly yields larger errors.

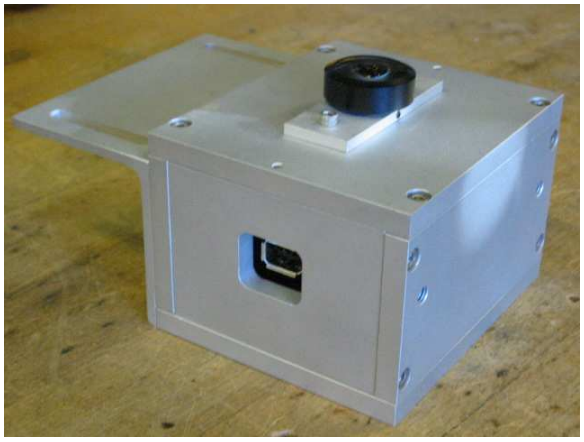


*Figure 1. Our sun tracker, in an enclosure with the fisheye lens visible on top.*

The operational principle of a sun tracker is straightforward. In general, the vectors pointing along the gravity direction and towards the sun provide two constraints on orientation. The only time the constraints are not independent is when the vectors are colinear, e.g. at noon on the equator on the equinox. With precise angular measurements the constraints are practically always independent, particularly when operating at mid-latitudes.

Given a location on a planetary body, the gravity vector can be computed with respect to the geoid in a global coordinate frame. With a clock, the direction toward the sun, or solar ephemeris, can be predicted. Software packages such as CSPICE[1] facilitate these computations. Optical sensors such as cameras and position sensing devices (PSD) can be used to measure the solar direction in sensor coordinates. Inclinometers can be used to measure the gravity direction. The sun tracker can provide good measurements as long as the sun is visible and the gravitational direction is within the range of the sensor. Estimating orientation is a matter of finding the rotation that aligns sensor frame observations to world frame vectors.

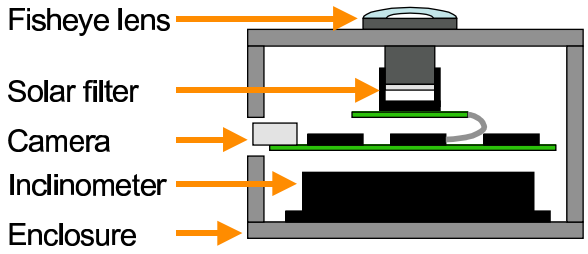



*Figure 2. Suntracker components.*

This paper describes a sun tracker built from simple, low power, off-the-shelf components and used on a planetary analog rover. The design, construction, mathematical modeling, calibration procedure, pose estimation technique, and quantitative results are presented.

## 2. SUN TRACKER DESIGN

Our sun tracker is shown in Figure 1, and is comprised of several components assembled as shown in Figure 2. The components are described in some detail below. The camera, optics, and inclinometer are all assembled in an aluminum enclosure and communicate to a host computer which reads data from the sensors simultaneously.

### 2.1. Camera and optics

The sun direction in the sensor coordinate frame is measured by a camera with appropriate optics. The sensor in our sun tracker is an off the shelf CCD imager, a Point Grey Dragonfly$^{TM}$. Position sensing devices (PSD), which provide an analog signal proportional to the location of the centroid of light on the detector, may also be used[11].

CCD imagers provide several advantages over PSD's without adding significant disadvantages. Under changing light levels, CCD pixel levels change without affecting the sun centroid, while some PSD's can suffer bias problems. A PSD only returns one measurement which can be corrupted by a reflection from a nearby specular surface, while a CCD image can be searched for more than one peak and rejected if a second reflection is detected. CCD measurements require more processing, but the computation to find single or multiple peaks and compute the centroid is trivial and can be done in one pass. A standard CCD camera offers 10 bits or more of position ($2^{10} = 1024$ pixels) even without subpixel estimation. Using a CCD or CMOS imager makes use of a modular system component. Our lab is familiar with CCD cameras, can easily add them to the firewire bus on our rover, and uses the same control and communication software which eases integration.

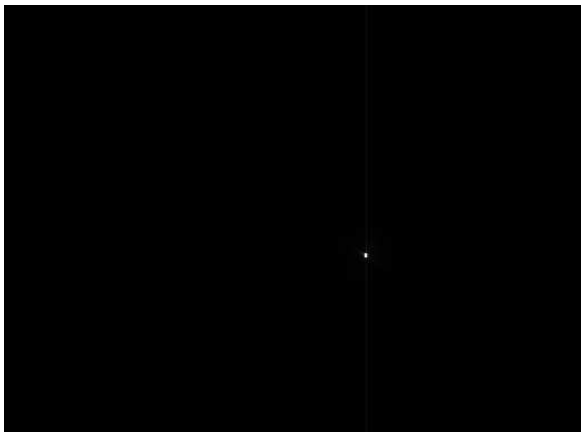


*Figure 3. Sample sun tracker image. The only non-dark region in the image is the projection of the sun.*

The optics filter and focus sunlight onto the detector. There is a tradeoff between total field of view (FOV) and angular resolution. Most available lenses conform to a few standard mount types. Our system uses an off the shelf fisheye microlens, the Omnitech ORIFL190-3, which provides a $190^o$ FOV. With an active imaging area of nearly 768 pixels, this corresponds to an average of about $1/4^o$ per pixel while the FOV covers slightly more than the whole sky. Note that the apparent size of the sun from Earth is roughly $1/2^o$.

Direct sunlight and bright scattered skylight cause saturation and blooming on the imager. To mitigate this, our sun tracker includes a neutral density filter which transmits a small fraction of light. The filter is a piece of AstroSolar$^{TM}$ mylar film from Baader Planetarium with transmission coefficient $T = 10^{-3.8} = 0.000158$, attached directly to the rear flange of the lens.

Figure 3 shows an example sun tracker image. Images of the sky through the ND filter and fisheye lens result in an image which is mostly black with a white spot corresponding to the projection of the sun on the image plane.

### 2.2. Inclinometer

The gravity direction is measured by an inclinometer inside the same enclosure. Orientation accuracy for the system is limited by the precision of the measurements of sun and gravity direction, so an accurate inclinometer is required. The inclinometer used in our sun tracker is a Crossbow CXTILT02EC, which has a reported accuracy of $0.1^o$ in roll and pitch. Because the power consumption is low and the voltage range matches the IEEE 1394 (firewire) spec, we power the device using 12V DC from the firewire cable to eliminate extra wiring. The inclinometer communicates through an RS-232 serial interface.

## 3. MODELING

There are many camera models used in computer vison. Probably the most popular is Tsai's model[9] for projection with the Brown-Conrady model of lens distortion[2]. The CAHVOR model[4] is popular within NASA. These two are largely the same and applicable to many real world cameras, but neither handles fisheye lenses well. Extensions add more distortion terms to model fisheye lenses along with the projective pinhole model[5, 6].

The model required for the sun tracker has different criteria. We are only interested in the relationship between image measurements and solar angles in the camera frame, which depend on solar ephemeris and the orientation of the sun tracker. We ignore the standard projective geometry approach and instead seek a mapping from solar angles directly to image plane coordinates.

We denote the Cartesian unit vector to the sun $\mathbf{x} = (x, y, z)^T$ and the corresponding spherical coordinates $\mathbf{\Theta} = (\theta, \phi)^T$ with azimuth $\theta$ and elevation $\phi$. We denote coordinate frames with subscripts, $w$ for world, $c$ for camera and $i$ for inclinometer, and rotations between frames with a subscript and superscript for the two related frames, e.g. $R_w^c$.

The solar ephemeris $\mathbf{\Theta_w}$ is rotated to spherical camera coordinates $\mathbf{\Theta_c}$ by the rotation $R_c^w$. This is carried out by converting $\mathbf{\Theta_w}$ to $\mathbf{x_w}$, rotating by $R_c^w$,

$$\mathbf{x_c} = R_c^w \mathbf{x_w} \tag{1}$$

and converting $\mathbf{x_c}$ to $\mathbf{\Theta_c}$. Then the measurement is

$$\mathbf{z} = \mathbf{h}(\mathbf{\Theta_c}, \mathbf{p}) \tag{2}$$

where $\mathbf{h}$ is the measurement model parameterized by $\mathbf{p}$, and $\mathbf{z_c} = (u_c, v_c)^T$ the coordinates of the sun on the image plane

$$\begin{aligned} \rho &= a_1\bar{\phi} + a_2\bar{\phi}^2 + a_3\bar{\phi}^3 + a_4\bar{\phi}^4 \\ u_c &= u_0 + \rho\cos(\theta_c) \\ v_c &= v_0 + \rho\sin(\theta_c) \end{aligned} \tag{3}$$

where $\bar{\phi} = \pi/2 - \phi$ is the complement of elevation. In terms of polar coordinates, the location of the sun on the image plane is given by the radial term $\rho$, modeled as a polynomial in elevation angle, and the angular term $\theta_c$, the solar azimuth. The image center is $(u_0, v_0)^T$, corresponding to $\phi = \pi/2$ or $\bar{\phi} = 0$.

The inclinometer reports roll $\alpha_i$ and pitch $\beta_i$ as the slope along the $x$ and $y$ axis of the sensor (not the roll and pitch corresponding to Euler RPY angles). The measurements can be modeled by

$$\begin{aligned} \alpha_i &= \text{asin}(R_w^i(3,2)) \\ \beta_i &= \text{asin}(R_w^i(3,1)) \end{aligned} \tag{4}$$

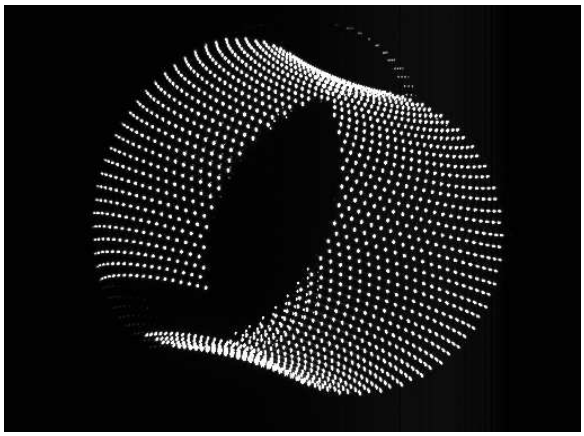


*Figure 4. Composite of 1600 images taken by rotating the sun tracker on a pan tilt unit (PTU). The images are used to calibrate the sensor.*

## 4. CALIBRATION

Calibration is the estimation of the paremeters $\mathbf{p} = (a_1, a_2, a_3, a_4, u_0, v_0, R_c^i)^T$. This presents a chicken and egg problem: neither the mapping in (3) nor the true orientation is known, so both must be estimated together. To disambiguate the problem, we put the sensor on a pan tilt unit (PTU) mounted on a tripod, and take a series of measurements at known pan and tilt angles. A composite of 1600 calibration images is shown in Figure 4. This data covers a significant portion of the image plane, providing strong constraints on the camera model.

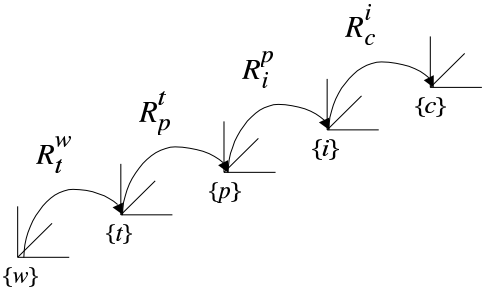



*Figure 5. Relationship between coordinate frames during calibration.*

The PTU provides relative orientation but absolute orientation is not known. Absolute orientation is the composition of rotations between the world *(w)*, tripod *(t)*, PTU *(p)*, inclinometer *(i)*, and camera *(c)* frames as shown in Figure 5. $R_t^p$ is assumed known since the PTU is accurate and controllable. The rotation between inclinometer and camera $R_i^c$ is a necessary part of the model. The other two rotations, $R_t^w$ and $R_p^i$ add 6 nuisance parameters to the 9 parameters of interest in $\mathbf{p}$. Then

$$\begin{aligned} R_w^i &= R_w^t R_t^p R_p^i \\ R_w^i &= R_w^t R_t^p R_p^i R_i^c \end{aligned} \tag{5}$$

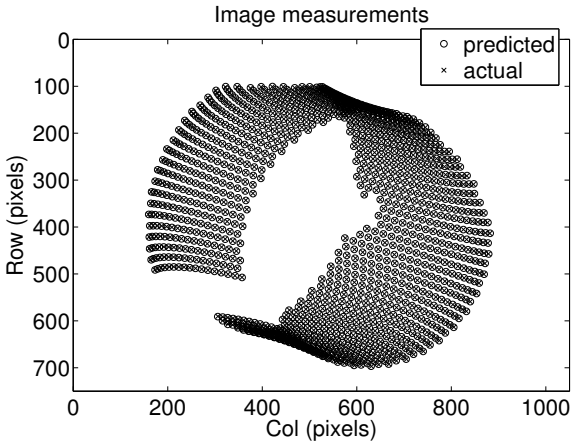



*Figure 6. Predicted vs. actual image measurements for the data in Figure 4*

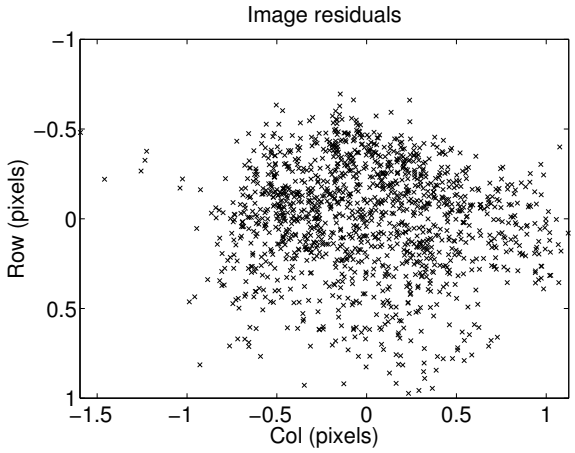



*Figure 8. Image measurement residuals after calibration are on the order of 0.25 pixels RMS.*

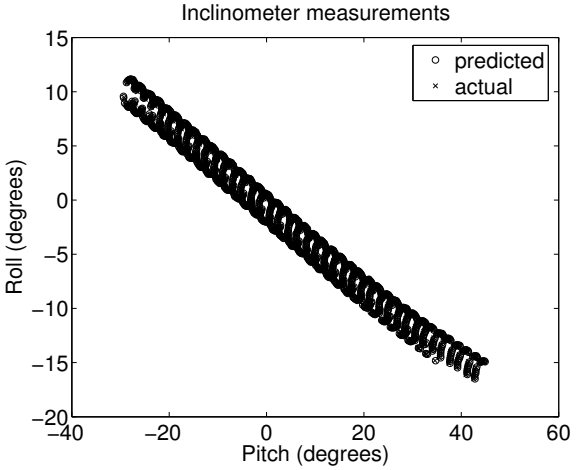



*Figure 7. Predicted vs. actual inclinometer measurements for the data in Figure 4*

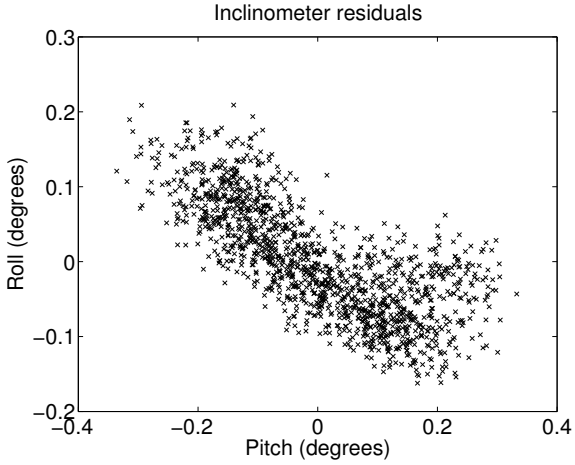



*Figure 9. Inclinometer measurement residuals after calibration are on the order of 0.1 degrees RMS.*

Calibration is done by fitting the 15 parameters in $\mathbf{p}$ using Levenberg-Marquardt[7]. We use numerical differentiation to simplify implementation and make it easier to experiment with sensor models. Since calibration is done offline and verified, there is little risk of being stuck with a poor solution. To improve convergence, optimization starts with a lower order model which assumes that $R_w^t$, $R_p^i$ and $R_i^c$ are yaw only, and that $a_2$, $a_3$ and $a_4$ are zero, (linear radial polynomial). This reduces the problem to 6 DOF. When the initial model converges, the full 15 parameter model is initialized with the low order solution and then optimized.

Figure 6 and 7 show the predicted and actual measurements for the camera and inclinometer for an example calibration. Figure 8 and 9 show the residuals between predicted and actual measurements for the camera and inclinometer. Note that the image residuals are on the order of 0.25 pixel RMS, and the inclinometer residuals are on the order of $0.1^o$ RMS which matches the manufacturer's specification. Figure 10 shows the convergence of the root mean square error of the cost function during the two step calibration process.

## 5. ESTIMATING ORIENTATION

The standard approach to estimating orientation for sun trackers is to compute the unit vector towards the sun by inverting the camera model, rotate the vector according to the roll and pitch estimate from an accelerometer or inclinometer, and then simply subtract the azimuth direction of the sun vector from the sensor observation from the solar azimuth given by ephemeris[8]. This method is computationally simple, but treats the two pieces of information differently and does not offer an internal check for solution quality since each of the two steps is only exactly properly constrained. That is, the roll and

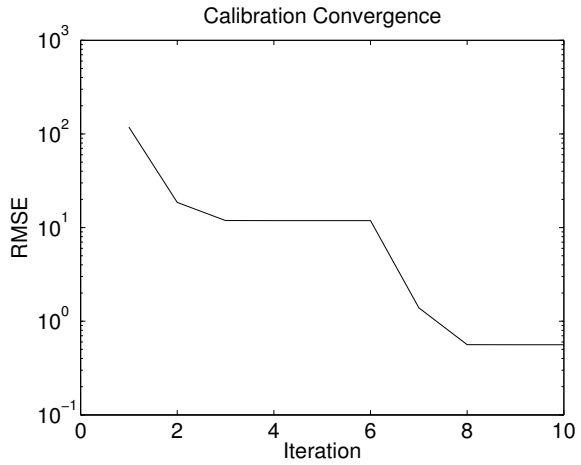



*Figure 10. Convergence of residual between predicted and actual measurements during calibration. Iterations 1-6 show convergence of the low order model, and iterations 7-10 the full model.*

pitch measurements are used exactly to rotate the observed sun vector, the resulting vector is then reduced to a one dimensional observation, and that one dimensional constraint is used to compute the yaw estimate.

Our method instead seeks to find the orientation such that the predicted measurements best match the actual measurements in a weighted least squares sense. This is again a nonlinear optimization problem: find the orientation $q$ such that the cost function

$$J = (\mathbf{z} - \mathbf{h}(q))^T \mathbf{\Sigma^{-1}} (\mathbf{z} - \mathbf{h}(q)) \qquad (6)$$

is minimized, where $\mathbf{z}$ is a vector containing the image coordinates of the sun location and the roll and pitch measurements from the inclinometer, and $\mathbf{\Sigma}$ is the measurement covariance, which is a diagonal matrix with $\sigma_{row} = \sigma_{col} = 0.25$ pixels and $\sigma_{roll} = \sigma_{pitch} = 0.1^o$. The measurement model $\mathbf{h}$ is the one described in Section 4 with the model parameters $\mathbf{p}$ fixed after calibration and the quaternion $q$ used to rotate the ephemeris to compute $\mathbf{\Theta_c}$. In reality, the measurement prediction requires the time and an approximate location so that ephemeris can be computed, but these are fixed for a given observation so we leave them out for notational convenience. To estimate orientation we seek the $q$ which minimizes (6).

Optimization problems involving rotations are sensitive to the parameterization of rotations, with unit quaternions earning their deserved status as a favorite method due to better convergence. But even searching over unit quaternions is a potentially messy nonlinear problem, so we estimate orientation in two steps. The first step is an initial estimate that does not necessarily minimize (6) but is computed in closed form. The second step optimizes the cost function in (6) directly.

Computing the initial orientation in closed form is done in a novel way. The camera and inclinometer observations can be thought of as measurements of the solar and gravitational directions in sensor coordinates. That is, the camera model in (3) can be inverted so that the location of the sun on the image plane can be converted to a unit vector pointing to the sun in camera coordinates, and the observed roll and pitch measurements can be used to compute a unit vector pointing along gravity in sensor coordinates. Solar ephemeris and the reference geoid provide unit vectors toward the sun and along gravity in world coordinates. These vectors can be aligned in closed form using part of the algorithm due to Umeyama[10]. The full algorithm is used to compute similarity transforms. A preprocessing step removes scale and translation, after which the algorithm finds the rotation between the corresponding vectors in closed form. Our application requires only the rotation fitting step to recover an estimate for $R_w^c$, and the two pairs of corresponding vectors are sufficient for Umeyama's algorithm. Computation is not prohibitive; the algorithm constructs a $3 \times 3$ matrix, computes its singular value decomposition (SVD), normalizes and checks the singular values in $S$ and then multiplies the $U$, $S$ and $V$ matrices to recover the rotation.

The second step in estimating orientation minimizes (6) directly using Levenberg-Marquardt[7]. The estimate from rotation fitting is used as an initial guess for the unit quaternion $q$ and L-M is iterated to convergence. The application of L-M to this problem is straightforward. Computational complexity is again not an issue. Iterations require the computation of $4 \times 4$ Jacobians (derivatives of four measurements with respect to four quaternion parameters) and the solution of a system of 4 linear equations. In practice only 3 to 6 iterations are required to minimize (6) sufficiently.

Provided the estimate is a global minimum, the orientation solution is optimal with respect to the cost function and observation. The initial guess can help ensure the solution is the global minimum since a closed form solution exists, but there is still a need to validate the estimate. The camera and inclinometer provide four constraints for three rotational degrees of freedom. This means that the solution quality can be evaluated after minimizing (6) by testing the residual against a chi-square with 1 degree of freedom. In practice, solutions which have a large residual are rejected since the best orientation estimate does not agree well with the measurement model and measurement uncertainties. This can happen if the camera sees a unique peak which is actually a strong reflection of the sun, or if the inclinometer measurement is compromised due to rover acceleration.

## 6. RESULTS

To test the accuracy of orientation estimates, we calibrate the sensor using a training set and then test orientation estimation with a test set. The calibration procedure is explained in Section 4 and orientation estimation is explained in Section 5.

Quantifying the errors in orientation estimates is difficult. The sensor orientation is estimated to within a fraction of a degree, but we lack a simple and independent method for measuring the ground truth orientation with similar precision.

To work around this, we use the unknown but fixed relationships in Figure 5 to help evaluate performance. It takes about an hour to collect 1600 images, stepping through 80 pan angles and 20 tilt angles. We can collect several data sets in a day and assume that $R_w^t$ and $R_p^i$, corresponding to the tripod orientation and the sensor mounting on the PTU, are constant for all data sets. Thus the fixed relationships in Figure 5 estimated using the calibration set can be used to evaluate the test set. The PTU provides $R_t^p$, so we can recover the orientation of the sensor by composing the rotations using (4). This orientation is considered to be the ground truth orientation.

Figure 11 shows the roll, pitch, and yaw estimation results for the test set. Deviations are very small compared to the range of orientations. Figure 12 shows histograms of the deviations between ground truth and estimated orientation parameters. For the test data set, the roll and pitch estimates match the ground truth to within about $0.15^o$ RMSE over the full range of orientations. This range covers most of the expected orientations for the rover, e.g. what is normally considered "safe". The yaw RMSE is around $0.4^o$. The error is within $1^o$ 89.75% of the time and is never worse than $2^o$. This is well within the goal of $3^o$ derived from the position estimation requirement of 5% distance travelled.

Results for position estimation for long traverses is beyond the scope of this paper, since position error depends on the fusion of multiple sources of information. These results will be reported elsewhere.

## 7. ACKNOWLEDGEMENTS

We would like to thank Hoang Vu at NASA Ames for design and assembly of the enclosure, Eric Park for help with electronics, Michael Wagner for supporting the integration and testing of the sun tracker on the Zoe rover at Carnegie Mellon and during LITA field campaigns, and Paul Tompkins, Clay Kunz, Randy Sargent, and Richard Volpe for many useful discussions. This work was funded by NASA's Astrobiology Science and Technology for the Exploration of the Planets (ASTEP) program.

## REFERENCES

1. http://naif.jpl.nasa.gov/naif/toolkit.html.

2. D.C. Brown. Lens distortion for close-range photogrammetry. *Photometric Engineering*, 37(8):855–866, 1971.

3. C. H. Cotter. *History of Nautical Astronomy*. Hollis & Carter, London, 1968.

4. D. B. Gennery. Least-squares camera calibration including lens distortion and automatic editing of calibration points. In H. Gruen and T. S. Huang, editors, *Calibration and Orientation of Cameras in Computer Vision*, volume 34 of *Springer Series in Information Sciences*, pages 123–136. Springer-Verlag, 2001.

5. D. B. Gennery. Generalized camera calibration including fish-eye lenses, 2002.

6. J. Kannala and S. Brandt. A generic camera calibration method for fish-eye lenses. In *International Conference on Pattern Recognition*, pages 10–13, 2004.

7. W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1988.

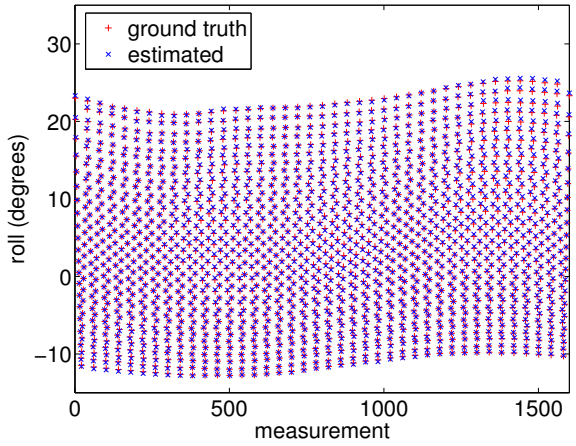
8. A. Trebi-Ollennu, T. Huntsberger, Y. Cheng, E. T. Baumgartner, B. Kennedy, and P. Schenker. Design and analysis of a sun sensor for planetary rover absolute heading detection. *IEEE Trans. on Robotics and Automation*, 17(6):939–947, Dec. 2001.

9. R. Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.

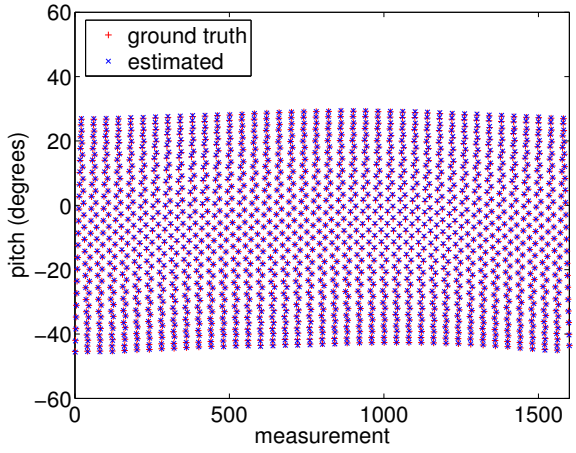
10. Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4), 1991.

11. R. Volpe. Mars rover navigation results using sun sensor heading determination. In *IEEE/RSJ International Conference on Intelligent Robot and Systems*, volume 1, pages 460–467, 1999.

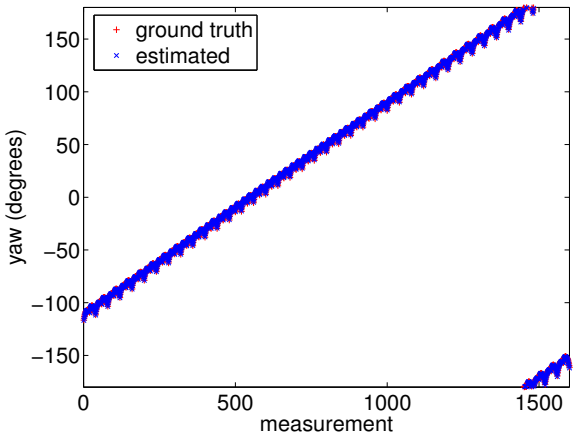
12. D. Wettergreen, N. Cabrol, J. Teza, P. Tompkins, C. Urmson, V. Verma, M.D. Wagner, and W.L. Whittaker. First experiments in the robotic investigation of life in the atacama desert of chile. In *International Conference on Robotics and Automation*, April 2005.




(a)

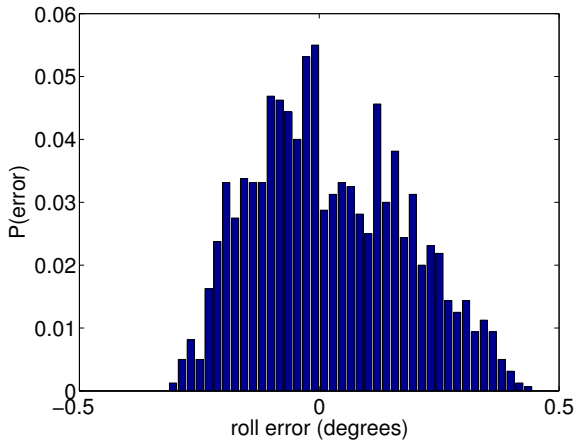



(b)

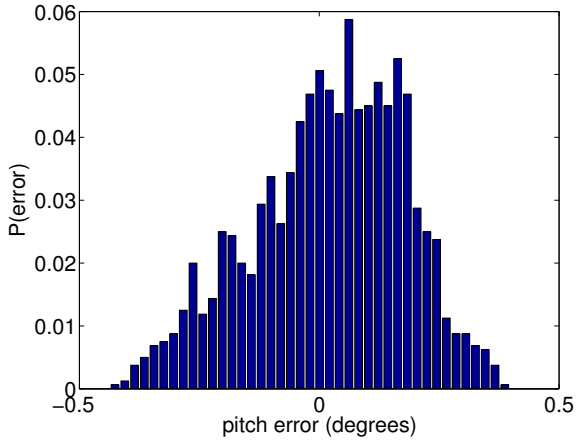



(c)

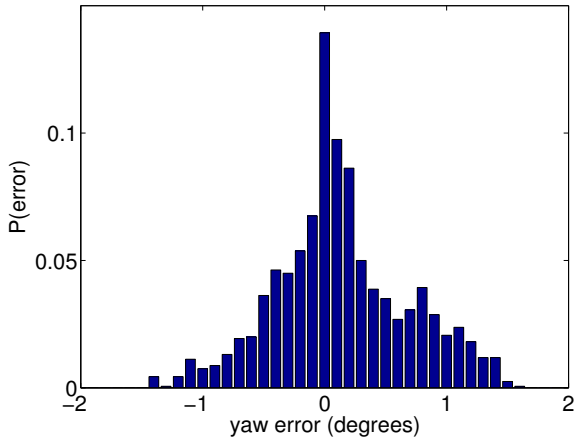
*Figure 11. Comparison of ground truth and estimated orientation. (a) Roll angle. (b) Pitch angle. (c) Yaw angle. Estimates are accurate over the full range of motion, which covers most of the range of orientations expected for the rover.*




(a)




(b)




(c)

*Figure 12. Histogram of estimation errors. (a) Roll angle, with RMSE around 0.15°. (b) Pitch angle, with RMSE around 0.15°. (c) Yaw angle, with RMSE around 0.4°. For the test set, yaw errors are typically below 1° and are always well below the desired accuracy of 3°.*